PR #26753 完整报告

sgl-project/sglang

[Bug] ngram verify: keep `batch.seq_lens_sum` in sync after accept

合并时间：2026-05-30 09:00

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26753

执行摘要

- 一句话：修复 ngram verify 后 seq_lens_sum 不同步导致 CUDA 越界
- 推荐动作：该 PR 是典型的隐蔽性 bug 修复，值得阅读以理解注意力后端对 seq_lens_sum 的依赖关系。对于关注推测解码稳定性的人员，建议合并此修复。

功能与动机

Ngram verify 在接受 tokens 后，batch.seq_lens 和 batch.seq_lens_cpu 已正确增加，但 batch.seq_lens_sum 未更新。Triton attention 后端在 eager 路径（batch size 超过 cuda-graph-max-bs 时）会从 forward_batch.seq_lens_sum 计算 kv_indices 缓冲区大小，导致下一次 verify 时缓冲区分配不足，create_flashinfer_kv_indices_triton 写入越界，表现为 CUDA error: illegal memory access。该模式与 dflash_info.py 中已有的同步逻辑一致（"Keep seq_lens_sum in sync; flashinfer indices updaters rely on this for buffer sizing"）。

实现拆解

1. 在文件 python/sglang/srt/speculative/ngram_info.py 的 verify 方法中，在更新 batch.seq_lens 和 batch.seq_lens_cpu 之后，增加一行代码：batch.seq_lens_sum += int(num_accept_tokens_cpu.sum())，确保 seq_lens_sum 与 seq_lens 保持同步。
2. 该变更位于 verify 方法末尾，紧接在 _free_cache 调用和 seq_lens_cpu 更新之后，返回之前。仅添加了 2 行代码（含注释），改动量极小。

关键文件：

- python/sglang/srt/speculative/ngram_info.py（模块 推测解码；类别 source；类型 core-logic）：核心变更文件，在 ngram verify 方法的正确路径末尾添加 seq_lens_sum 同步语句，修复隐蔽的缓冲区越界 bug。

关键符号：未识别

关键源码片段

python/sglang/srt/speculative/ngram_info.py

核心变更文件，在 ngram verify 方法的正确路径末尾添加 seq_lens_sum 同步语句，修复隐蔽的缓冲区越界 bug。

```
# python/sglang/srt/speculative/ngram_info.py (line 466-469)
```

```
# 在更新 batch.seq_lens 和 batch.seq_lens_cpu 后，
# 同步更新 batch.seq_lens_sum，因为注意力后端（如 Triton）
# 在 eager 模式下使用 seq_lens_sum 来分配 kv_indices 缓冲区大小。
batch.seq_lens.add_(self.num_accept_tokens)
batch.seq_lens_cpu.add_(num_accept_tokens_cpu)
# Keep seq_lens_sum in sync; attn backends size kv_indices from it.
batch.seq_lens_sum += int(num_accept_tokens_cpu.sum())
```

评论区精华

无 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。变更仅在 ngram verify 正确路径末尾添加一行整数加法，与已有的 seq_lens 和 seq_lens_cpu 更新逻辑完全一致，且与 flashinfer 后端保持模式一致。需要确保 num_accept_tokens_cpu.sum() 返回值类型与 seq_lens_sum 兼容（均为 Python int），当前代码使用 int() 转换保证了兼容性。
- 影响：直接影响 ngram 推测解码场景下 Triton attention 后端 eager 模式的正确性，修复了偶发的 CUDA illegal memory access 崩溃。对非 ngram 或使用其他 attention 后端的场景无影响。
- 风险标记：核心路径变更

关联脉络

- PR #26128 [core] Make spec_v2 seq_lens_cpu optional via backend needs_cpu_seq_lens; Triton opts out: 同样是关于推测解码中 seq_lens 同步的改进，展示了注意力后端对 seq_lens 数据的依赖模式。